User's guide to ChIP-Seq applications in the Vital-IT environment

1. Basics about the ChIP-Seq Tools

The ChIP-Seq software provides a set of tools performing common genome-wide ChIP-Seq analysis tasks, including positional correlation analysis, peak detection, and genome partitioning into signal-rich and signal-poor regions. These tools exist as stand-alone C programs and perform the following tasks:

I. Positional correlation analysis (chipcor),
II. Tag centering (chipcenter),
III. Signal peaks detection (chippeak),
IV. Identification of signal-enriched regions (chippart),
V. Feature extraction tool (chipscore).

Because the ChIP-Seq tools are primarily optimized for speed, they use their own compact format for ChIP-Seq data representation called SGA (Simplified Genome Annotation). SGA is a line-oriented, tab-delimited plain text format with the following five obligatory fields per line:

I. Sequence ID (char string)
II. Feature (char string)
III. Sequence Position (integer)
IV. Strand (+/- or 0)
V. Tag Counts (integer)

An additional field may be added containing application-specific information used by other programs. In the case of ChIP-Seq data, an SGA file represents the genome-wide tag count distributions from one or several experiments. However, the format can also represent a large variety of other types of genomic data, including derived features such as peaks extracted from ChIP-Seq data, or genome annotations such as promoters. Any type of genomic feature that can be projected to a single base on a chromosome can be represented in SGA format.

The *sequence* field typically identifies a chromosome. The software does not impose any naming convention. However, if you merge data from different experiments, the same names must be used for the same chromosomes. The public data files access by the ChIP-Seq server menu use NCBI/RefSeq accession numbers as sequence/chromosome identifiers. Chromosome names as used by the UCSC genome browser constitute another *de facto* standard and are readily converted into NCBI/RefSeq accession numbers. However, keep in mind that chromosome names ambiguous unless the corresponding assembly is indicated. Do not mix chromosome names from different assemblies otherwise you will get wrong results.

The f*eature* field contains a short code which identifies an experiment. It often corresponds to the name of the molecular target of a ChIP-Seq experiment. Its function is

to distinguish data lines relating to different experiments that were merged into a single file. The *position* field contains the position within the sequence. The *strand* field indicates the strand to which the feature has been mapped. The SGA format distinguishes between "oriented" features that occur either on the plus or on the minus strand of the chromosome sequence, and "un-oriented" features which cannot be assigned to one or the other strand. Peaks from a ChIP-Seq experiment, for instance, constitute an example of an un-oriented feature. Un-oriented features are identified by a 0 (zero) in field 4. The *counts* field contains the number of reads that have been mapped to a specific base position on the chromosome.

An example of an SGA-formatted file is shown here below:

```
NC_000001.9     H3K4me3     4794    +       1
NC_000001.9     H3K4me3     6090    +       1
NC_000001.9     H3K4me3     6099    +       1
NC_000001.9     H3K4me3     6655    +       1
NC_000001.9     H3K4me3     18453   -       1
NC_000001.9     H3K4me3     19285   +       1
NC_000001.9     H3K4me3     44529   +       1
NC_000001.9     H3K4me3     46333   +       1
NC_000001.9     H3K4me3     46349   -       1
```

Chip-Seq programs require SGA files to be sorted by sequence name, position, and strand. In a UNIX environment, the command to properly sort SGA files is the following:

sort -s -k1,1 -k3,3n -k4,4 <SGA file>

2. Using ChIP-Seq Tools on the Vital-IT platforms

To use the ChIP-Seq tools from the command line, login to a Vital-IT machine, *e.g.*

ssh -l gambrosi frt.el.vital-it.ch

The ChIP-Seq programs are installed on Vital-IT in the directory */mnt/local/bin*. Make sure that this directory is in your path. To add it to your PATH environment variable from a bash shell. type:

PATH=$PATH:/mnt/local/bin; export PATH

You invoke the programs by simply typing the program name, i.e.: '*chipcor', 'chipcenter', 'chippeak', 'chippart', 'chipscore'*.

Here is an example of using the ChIP-Seq correlation program *chipcor*:

chipcor -A "CTCF +" -B "CTCF -" -b -1000 -e 1000 -w 1 -c 1 -n 1 CTCF.sga > CTCF.out

where:

CTCF.sga	is the input file containing the list of mapped CTCF tags.
-A "CTCF +"	is the reference feature (CTCF plus strand)
-B "CTCF −"	is the target feature
-b	is beginning of the range considered in the output histogram
-e	is the end of the range
-w	is the window width
-c	is the count cut-off value
-n	is the normalization mode (1 means "count density")
CTCF.out	is the output file containing the histogram values in text format

The input file for this example can be found at:

/db/mga/hg18/barski07/CTCF.sga

Documentation for ChIP-Seq programs is provided via UNIX- style man page, type *e.g.*:

man chipcor

Short usage instructions can also be obtained by simply typing the name of the program without any options or arguments.

In the above example, the reference and target features were contained in the same input file. This is not always the case. In the following example, we will analyze the distribution of CTCF peaks in mouse embryonic stem cells relative to transcription start sites from the Eukaryotic Promoter Database EPD. The input data for this analysis are provided in two separate files which can be found at:

/db/mga/mm9/chen08/ES_CTCF_peaks.sga
/db/mga/mm9/epd/Mm_EPDnew_001_mm9.sga

These files need first to be merged before they can be processed by *chipcor*.

```
sort -s -m -k1,1 -k3,3n -k4,4 Mm_EPDnew_001_mm9.sga ES_CTCF_peaks.sga \
 > merged.sga
chipcor -A "TSS" -B "CTCF_P" -b -2500 -e 2500 -w 50 -c 1 -n 1 -o merged.sga \
 > CTCF_peaks.out
```

The public data which are accessible via the ChIP-Seq server menu are stored in the so-called MGA (Mass Genome Annotation) data repository located in */db/mga*. All data in the MGA repository are provided in SGA format. At the first hierarchical level, the *mga* root directory is split into sub-directories corresponding to genome assemblies (*e.g.* hg18, mm9), at the second level according to the data series (*e.g.* hg18/barski07). A third level is used for ENCODE data sets (*e.g.* hg18/encode/GSE25416). A data series sub-directory typically contains data from one publication and/or one series in GEO (GSE entry).

3. Auxiliary Tools for data reformatting

We also provide a series of auxiliary tools (most of them perl scripts) that can be used to perform format conversion tasks. These perl scripts are installed in */mnt/common/perl/*. However, some of the third-party reformatting tools are not installed on all Vital-IT machines. We therefore recommend that you carry out all reformatting tasks on the ChIP-Seq server machines directly.

Most of the ChIP-seq data sets come in BAM or BED formats. If you have BAM files, and you need to convert them to SGA format, the steps to follow are:

1. Log into the ChIP-Seq auxiliary server machine on Vital-IT:

   ssh -l username ccg-serv02.vital-it.ch

2. Make sure that the directories /home/local/bin and /home/local/perl are in your PATH environment variable, *e.g.* on a bash shell command line type:

   PATH=$PATH:/home/local/bin:/home/local/perl; export PATH

3. Find out which genome assembly has been used to generate the BAM files, and make sure that the chromosome names agree with the naming scheme of the UCSC genome browser. Then use the following type of command.

bamToBed -i reads.bam | bed2sga.pl -f <feature> -s <species> | sort -s -k1,1 -k3,3n -k4,4 | compactsga > reads.sga

The program *bamToBed* belongs to the **BEDTools** package, a suite of utilities for comparing genomic features that can be installed from

http://code.google.com/p/bedtools.

bamToBed converts BAM alignments to BED format.The '*bed2sga.pl'* tool is a perl script that converts BED files to SGA format. To find instructions how to use it, type:

bed2sga.pl -h

SGA format requires a *feature* field (the second field) containing a character string that defines the genomic feature represented by the file. The BED file does not have an equivalent field. You therefore need to supply a feature for the conversion via the *-f* option. The *species* is the name used by UCSC for the genome assembly to which the reads have been mapped (*e.g.* hg18, mm9, dm3, etc.). *bed2sga.pl* will automatically convert UCSC chromosome names into corresponding NCBI/RefSeq accession numbers.

The reasons why sorting is required have been explained before. The program '*compactsga'* is a C program which merge lines corresponding to the same genome position (identical sequence name, position and strand). For instance, the two input lines

```
NC_000001.9     H3K4me3     5011    +       1
NC_000001.9     H3K4me3     5011    +       1
```

would be replaced by the following single line:

```
NC_000001.9     H3K4me3     5011    +       2
```

Compacting sga files saves space and reduces program execution time, but unlike sorting is not formally required by the ChIP-Seq tools.

The *bed2sga.pl* has two basic modes of operations, *centered* and *oriented*. In the *centered* mode, the midpoint between the *start* and *end* position from the BED line (2nd and 3rd field) will be used as position. In the *oriented* mode, the conversion depends on the *strand* indicated in the BED file (6th field). If the strand is + then the value of the *start* field will be incremented by one and used as position in the SGA file. If the strand is −, the value of the *end* field will be used as position in the SGA file. Incrementing the start position in *oriented* mode is necessary because the BED format has a "zero-based" numbering system for chromosomal regions whereas SGA has a 1-based numbering system. The behavior of the two conversion modes is illustrated by the following example:

BED input:

```
chr1   100000266  100000291   .  0   +
chr1   100000383  100000408   .  0   -
```

SGA output, centered mode:

```
NC_000001.9  CHIPSEQ  100000278  +   1
NC_000001.9  CHIPSEQ  100000395  -   1
```

SGA output, oriented mode:

```
NC_000001.9  CHIPSEQ  100000267  +   1
NC_000001.9  CHIPSEQ  100000408  -   1
```

By the default, the conversion mode depends on the contents of the BED file. If the strand field (which is optional in BED) is presented, the conversion will be done in *oriented* mode. Otherwise, the conversion will be done in centered mode, and the strand field will be set to zero. However, centered mode can be forced by the command the line option *–c*.

The most common command pipeline to perform BED-to-SGA conversion is the following:

bed2sga.pl -f <feature> -s <species> reads.bed | sort -s -k1,1 -k3,3n -k4,4 | compactsga > reads.sga

There are several additional reformatting programs that may be useful in certain situation. The C program *featreplace* replaces all feature names in an SGA file by a new name:

featreplace –f <feature> old.sga > new.sga

The liftOver program from UCSC can be used to convert chromosomal coordinates in a BED file from one genome assembly to another, *e.g.*:

liftOver input.bed /db/liftOver/mm8ToMm9.over.chain.gz output.bed trash

Note that liftOver chain files can be found in */db/liftOver*.

Additional conversion scripts include:

sga2bed.pl	SGA to BED
sga2wigFS.pl	SGA to WIG (wiggle) fixedStep format
sga2wigVS.pl	SGA to WIG variableStep format

The latter two programs produce custom track files that can be uploaded to the UCSC genome browser for visualization.

4. Working on the Web with your own server-resident data

If you would like to use the ChIP-Seq server web interface (http://ccg.vital-it.ch/chipseq) with your own large data files, we strongly recommend that you store your data directly on the web server machine. You can then access your file via the input menu option "Server -resident SGA Files by Filename". By choosing this data access mode, you can avoid transfer of large files over the network.

To be able to store your data directly on the server, proceed as follows.

1) Send an email request to ccg@mail.isrec.isb-sib.ch indicating your Vital-It *username.* We will then create a repository for your data on the server machine ccg-serv01.vital-it.ch in */data/scratch/username* to which you will have read/write access. Note that files in this area will be permanently stored but not backed-up.
2) Create a sub-directory corresponding to the genome assembly of your data. The name of the subdirectory should conform to the UCSC nomenclature (*e.g*. hg18, hg19, mm9, dm3, ce6, sacCer3)
3) Transfer your data to the appropriate subdirectory on ccg-serv01.vital-it.ch. Note that only files in SGA format can be read from this location.
4) On the Web interface, select your file via the menu "Server-resident SGA Files by Filename", by typing

   username/assembly/myreads.sga (e.g. *gambrosi/hg18/All_histones.sga*)

   into the text area.